

Synchronous Data Link Control and Derivatives

Background

IBM developed the Synchronous Data Link Control (SDLC) protocol in the mid-1970s for use in Systems Network Architecture (SNA) environments. SDLC was the first link-layer protocol based on synchronous, bit-oriented operation. This chapter provides a summary of SDLC's basic operational characteristics and outlines several derivative protocols.

After developing SDLC, IBM submitted it to various standards committees. The International Organization for Standardization (ISO) modified SDLC to create the *High-Level Data Link Control* (HDLC) protocol. The International Telecommunication Union–Telecommunication Standardization Sector (ITU-T;) (formerly CCITT) subsequently modified HDLC to create *Link Access Procedure* (LAP), and then *Link Access Procedure, Balanced* (LAPB). The Institute of Electrical and Electronic Engineers (IEEE) modified HDLC to create *IEEE 802.2*. Each of these protocols has become important in its own domain, but SDLC remains the primary SNA link-layer protocol for WAN links.

SDLC Types and Topologies

SDLC supports a variety of link types and topologies. It can be used with point-to-point and multipoint links, bounded and unbounded media, half-duplex and full-duplex transmission facilities, and circuit-switched and packet-switched networks.

- SDLC identifies two types of network nodes: *primary* and *secondary*. *Primary* nodes control the operation of other stations, called *secondaries*. The primary polls the secondaries in a predetermined order, and secondaries can then transmit if they have outgoing data. The primary also sets up and tears down links and manages the link while it is operational. *Secondary* nodes are controlled by a primary, which means that secondaries can send information to the primary only if the primary grants permission.

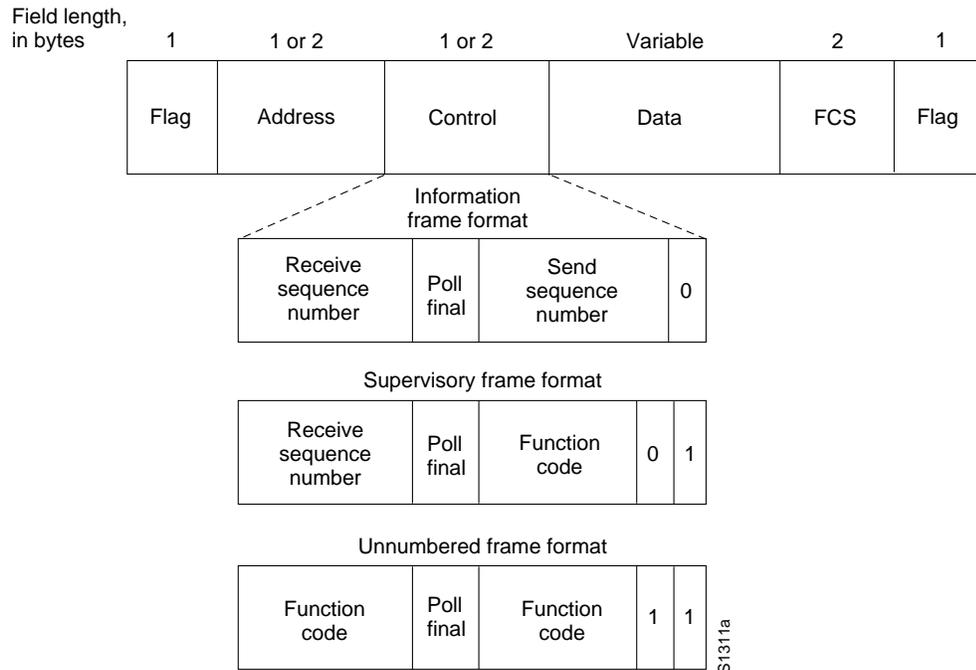
SDLC primaries and secondaries can be connected in four basic configurations:

- *Point-to-point*—Involves only two nodes, one primary and one secondary.
- *Multipoint*—Involves one primary and multiple secondaries.
- *Loop*—Involves a loop topology, with the primary connected to the first and last secondaries. Intermediate secondaries pass messages through one another as they respond to the requests of the primary.
- *Hub go-ahead*—Involves an inbound and an outbound channel. The primary uses the outbound channel to communicate with the secondaries. The secondaries use the inbound channel to communicate with the primary. The inbound channel is daisy-chained back to the primary through each secondary.

SDLC Frame Format

The SDLC frame is shown in Figure 16-1.

Figure 16-1 Six fields comprise the SDLC frame.



The following descriptions summarize the fields illustrated in Figure 16-1:

- *Flag*—Initiates and terminates error checking.
- *Address*—Contains the SDLC address of the secondary station, which indicates whether the frame comes from the primary or secondary. This address can contain a specific address, a group address, or a broadcast address. A primary is either a communication source or a destination, which eliminates the need to include the address of the primary.
- *Control*—Employs three different formats, depending on the type of SDLC frame used:
 - *Information (I) frame*: Carries upper-layer information and some control information. This frame sends and receives sequence numbers, and the poll final (P/F) bit performs flow and error control. The send- *sequence number* refers to the number of the frame to be sent next. The receive-*sequence number* provides the number of the frame to be received next. Both sender and receiver maintain send- and receive-sequence numbers.

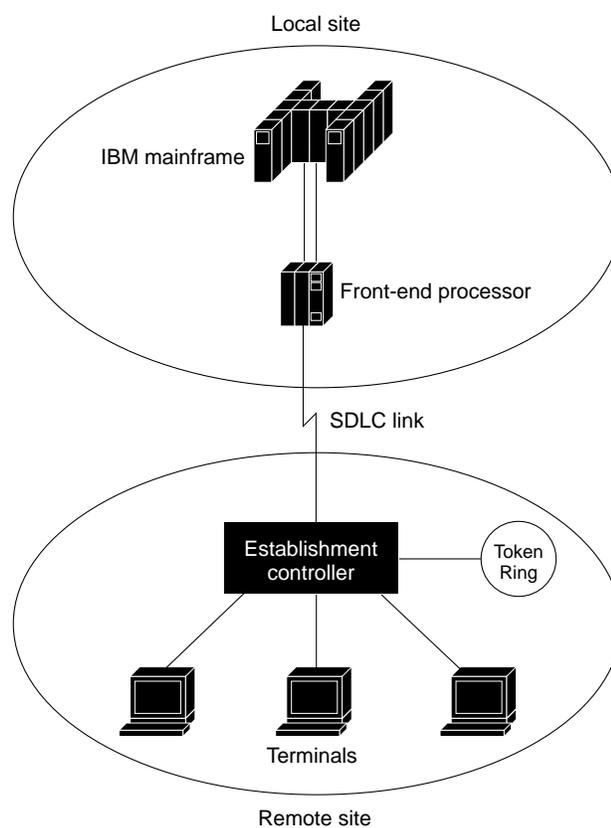
A primary station uses the P/F bit to tell the secondary whether it requires an immediate response. A secondary station uses the P/F bit to tell the primary whether the current frame is the last in its current response.
 - *Supervisory (S) frame*: Provides control information. An S frame can request and suspend transmission, reports on status, and acknowledge receipt of I frames. S frames do not have an information field.
 - *Unnumbered (U) frame*: Supports control purposes and is not sequenced. A U frame can be used to initialize secondaries. Depending on the function of the U unnumbered frame, its control field is 1 or 2 bytes. Some U unnumbered frames have an information field.

- *Data*—Contains a path information unit (PIU) or exchange identification (XID) information.

Frame Check Sequence (FCS)—Precedes the ending flag delimiter and is usually a *cyclic redundancy check* (CRC) calculation remainder. The CRC calculation is redone in the receiver. If the result differs from the value in the original frame, an error is assumed.

A typical SDLC-based network configuration is shown in Figure 16-2. As illustrated, an IBM establishment controller (formerly called a cluster controller) in a remote site connects to dumb terminals and to a Token Ring network. In a local site, an IBM host connects (via channel-attached techniques) to an IBM front-end processor (FEP), which also can have links to local Token Ring LANs and an SNA backbone. The two sites are connected through an SDLC-based 56-kbps leased line.

Figure 16-2 An SDLC line links local and remote sites over a serial line.



Derivative Protocols

Despite the fact that it omits several features used in SDLC, HDLC is generally considered to be a compatible superset of SDLC. LAP is a subset of HDLC and was created to ensure ongoing compatibility with HDLC, which had been modified in the early 1980s. IEEE 802.2 is a modification of HDLC for LAN environments. Qualified Logical Link Control (QLLC) is a link-layer protocol defined by IBM that enables SNA data to be transported across X.25 networks.

High-Level Data Link Control (HDLC)

HDLC shares the frame format of SDLC, and HDLC fields provide the same functionality as those in SDLC. Also, as in SDLC, HDLC supports synchronous, full-duplex operation.

HDLC differs from SDLC in several minor ways, however. First, HDLC has an option for a 32-bit checksum. Also unlike SDLC, HDLC does not support the loop or hub go-ahead configurations.

The major difference between HDLC and SDLC is that SDLC supports only one transfer mode, whereas HDLC supports three:

- *Normal response mode (NRM)*—This transfer mode is also used by SDLC. In this mode, secondaries cannot communicate with a primary until the primary has given permission.
- *Asynchronous response mode (ARM)*—This transfer mode enables secondaries to initiate communication with a primary without receiving permission.
- *Asynchronous balanced mode (ABM)*—ABM introduces the *combined* node, which can act as a primary or a secondary, depending on the situation. All ABM communication occurs between multiple combined nodes. In ABM environments, any combined station can initiate data transmission without permission from any other station.

Link-Access Procedure, Balanced (LAPB)

LAPB is best known for its presence in the X.25 protocol stack. LAPB shares the same frame format, frame types, and field functions as SDLC and HDLC. Unlike either of these, however, LAPB is restricted to the ABM transfer mode and is appropriate only for combined stations. Also, LAPB circuits can be established by either the data terminal equipment (DTE) or the data circuit-terminating equipment (DCE). The station initiating the call is determined to be the primary, and the responding station is the secondary. Finally, LAPB use of the P/F bit is somewhat different from that of the other protocols. For details on LAPB, see Chapter 17, “X.25.”

IEEE 802.2

IEEE 802.2 is often referred to as the *Logical Link Control (LLC)*. It is extremely popular in LAN environments, where it interoperates with protocols such as *IEEE 802.3*, *IEEE 802.4*, and *IEEE 802.5*. IEEE 802.2 offers three types of service.

Type 1 provides unacknowledged connectionless service, which means that LLC Type 1 does not confirm data transfers. Because many upper-layer protocols, such as *Transmission Control Protocol/Internet Protocol (TCP/IP)*, offer reliable data transfer that can compensate for unreliable lower-layer protocols, Type 1 is a commonly used service.

Type 2 provides connection-oriented service. LLC Type 2 (often called *LLC2*) service establishes logical connections between sender and receiver and is therefore connection oriented. LLC2 acknowledges data upon receipt and is used in IBM communication systems.

Type 3 provides acknowledged connectionless service. Although LLC Type 3 service supports acknowledged data transfer, it does not establish logical connections. As a compromise between the other two LLC services, LLC Type 3 is useful in factory-automation environments where error detection is important but context storage space (for virtual circuits) is extremely limited.

End stations can support multiple LLC service types. A Class I device supports only Type 1 service. A Class II device supports both Type 1 and Type 2 service. Class III devices support both Type 1 and Type 3 services, and Class IV devices support all three types of service.

Upper-layer processes use IEEE 802.2 services through *service access points* (SAPs). The IEEE 802.2 header begins with a *destination service access point* (DSAP) field, which identifies the receiving upper-layer process. In other words, after the receiving node's IEEE 802.2 implementation completes its processing, the upper-layer process identified in the DSAP field receives the remaining data. Following the DSAP address is the *source service access point* (SSAP) address, which identifies the sending upper-layer process.

Qualified Logical-Link Control (QLLC)

QLLC provides the data link control capabilities that are required to transport SNA data across X.25 networks. QLLC and X.25 replace SDLC in the SNA protocol stack. QLLC uses the packet-level layer (Layer 3) of the X.25 protocol stack. To indicate that a Layer 3 X.25 packet must be handled by QLLC, a special bit called the *qualifier bit*, in the *general format identifier* (GFI) of the Layer 3 X.25 packet-level header is set to one. The SNA data is carried as user data in Layer 3 X.25 packets. For more information about the X.25 protocol stack, see Chapter 17, "X.25."

